

Dia da Aceitação Universal Uruguai - 26 de maio de 2025

Sessão Teórica: Conceitos-chave da Aceitação Universal

Nicolás Antoniello – ICANN

Carlos Martínez – LACNIC

Sessão Teórica: Conceitos-chave da Aceitação Universal

Nicolás Antoniello – ICANN
Carlos Martínez – LACNIC

Carlos Martínez: Bem, estamos aqui para comemorar um pouco, porque ouvimos uma série de apresentações ao longo da manhã discutindo todos os benefícios e a conveniência de ter uma internet multilíngue, domínios multicaracteres, etc. Ou

porque alguém tem que fazer isso, certo? Então, vamos mostrar, eu diria, apenas arranhar a superfície da complexidade que está por trás disso.

Eu, e reconheço que sou historicamente cético em relação ao assunto, achava que o problema era exagerado até começar a estudá-lo um pouco, e a complexidade disso é incrível e a minimizamos porque na realidade temos, na língua que falamos, cinco, seis e talvez alguns dos mais antigos lembrem-se do sétimo caractere especial, que são os cinco acentos, o ñ e o trema . É tudo o que temos. E o trema , acho que meu filho não sabe o que é, por exemplo.

Outros conjuntos de caracteres têm complexidades muito, muito, muito, muito piores, como conjuntos de caracteres em que os caracteres são diferenciados em versões estreitas e amplas. A mesma letra, mas é larga e estreita e significa coisas diferentes. Cada uma dessas coisas tem que ser codificada de alguma forma.

Então, bem, hoje, entre esta manhã e esta tarde, vamos mostrar a vocês... Falar um pouco sobre essa complexidade e mostrar como abordar e fazer algumas coisas com um servidor de e-mail e um domínio. Nico, você que supostamente sabe de tudo isso. Avançar.

Nicolás Antoniello : Bom, eu estava bebendo um pouco de Coca-Cola porque meus colegas da ICANN estavam me dizendo, não coma meu alfajor de amido de milho antes de falar, digamos, você vai começar a... É complicado, não coma meu alfajor de amido de milho antes de falar. Moral da história de hoje.

Bom. Como Cristian disse mais cedo hoje, faça perguntas, interrompa, faça o que quiser. Novamente, com esses tipos de apresentações, devemos já saber o que estamos apresentando. Então, essas são mais para você, para esclarecer quaisquer dúvidas. E eu e o Carlos gostamos de fazer essas apresentações que têm uma... Aliás, deixamos para ele uma porcentagem grande de improvisação. Porque acho que isso os torna mais dinâmicos. Mas uma grande parte disso também é você fazendo perguntas e nós interagindo.

A apresentação em si será um guia, então talvez nem cheguemos ao final dela. E vai depender um pouco de como a conversa vai acontecer. Sim, vai ficar um pouco mais técnico, só um pouco, nada mais. E à tarde vai ficar um pouco mais técnico do que agora. Mas a ideia também é que todos possam segui-lo, mesmo que não sejam especialistas técnicos. Sim?

Bom. Isso já foi dito várias vezes mais cedo hoje. Então será uma revisão rápida. O que é aceitação universal? Basicamente, suporte à escrita em todos os sistemas, aplicativos, em outras linguagens de escrita além do código ASCII. Então basicamente não é inglês. Sim? E o ASCII nem sequer abrange todo o inglês, por assim dizer, porque há muitos caracteres, separadores, pontuação, etc., do inglês, que nem sequer são abrangidos pelo ASCII. Isto não é inglês. Este é um código ASCII, que é ainda mais restritivo que o inglês.

Essa é a versão original do DNS, o sistema de nomes de domínio. Ele suportava apenas caracteres ASCII, codificação ASCII em nomes de domínio. E e-mail também, apenas caracteres ASCII para nomes de caixa de correio. E bem, obviamente, do lado do DNS, o que vem antes do sinal @ , que é o nome da caixa, e o que vem depois do sinal @ . Ou seja, à esquerda do sinal @ está o nome da caixa, à direita do sinal @ está o nome do domínio. Nenhum deles, na versão original desses padrões, podia ser escrito em caracteres diferentes de ASCII. Sim?

E vou dar um spoiler . Vou dar uma prévia de algo que está no final da apresentação. Porque não é só e-mail. O e-mail faz parte do que é universalmente aceito. Na verdade, nessa parte de aceitação universal, você verá uma sigla em inglês, que acredito ser algo como Endereço de E-mail Internacionalizado , IEA. O IEAS é a versão por e-mail da aceitação universal, por assim dizer.

Mas, novamente, não adianta ter todos os servidores de e-mail aceitando. E quando eu vou pedir para um usuário criar uma conta de usuário para acessar meu site, criar uma conta de usuário e escolher um nome de usuário e uma senha, então eu digo para ele escolher, mas na realidade é mentira, eu não deixo ele escolher muito porque ele só pode usar o endereço de e-mail dele com o nome de usuário e ele pode escolher a senha, felizmente. Mas quando você vai escrever o nome do e-mail, o endereço de e-mail, se meu e-mail fosse nicolas @, não sei, internet.com.uy, acho que eles não existem mais. Na verdade, o meu era nicolás@internet.com.uy sem acento, porque eu não sabia colocar acento. Agora eu coloco o til , mas quando eu vou para a página da web, ele não aceita, porque o parser , o software, digamos, que foi usado para ver se o que eu estou colocando ali é um e-mail, quando ele vê algo que não é ASCII, ele diz, como é que isso não é um e-mail? O que você quer dizer com "não é um e-mail"? Está um pouco desatualizado.

Trata-se de atualizar todos os sites e todos os aplicativos em geral para oferecer suporte a múltiplas gravações, por assim dizer. Bem, isso é aceitação universal.

Por que isso importa? Também já dissemos: apoiar uma internet diversificada e multilíngue, permitindo maior concorrência, inovação e escolha do consumidor, criando oportunidades de negócios, só isso. Do meu ponto de vista pessoal, é bem relativo, mas também é importante e é um dos motores de tudo no mundo, certo? E, bem, ajudar governos a desenvolver políticas e também, não está explicitamente declarado aqui, mas está implícito, digamos, há muita manutenção da identidade, preservação da identidade cultural, certo?

Há pessoas que não sabem escrever em outra língua. Há pessoas que não sabem escrever em nenhuma outra língua além da sua e não conhecem nenhuma outra língua. E há outros que, com todo o direito do mundo, sabem três, quatro ou cinco línguas, muito mais línguas do que eu posso saber, mas não querem escrever em outra língua porque querem preservar sua identidade. Então isso também é sobre isso, certo?

Bom. Então, vamos falar um pouco sobre o que é aceitação universal, com mais detalhes, da perspectiva de um nome de domínio, ou seja, da perspectiva de um sistema de nomes de domínio, e o que é aceitação universal da perspectiva de um endereço de e-mail, ok?

Levante a mão se você não sabe o que é o sistema de nomes de domínio. Não se preocupe, isso não é um teste e você não precisa saber tudo sobre o sistema de nomes de domínio. Todo mundo sabe o que é. Pelo menos todo mundo acha que sabe o que é. E então, quando você pergunta a ele, ele diz: Eu sei o que é, mas não consigo explicar. Isso acontece, acontece com todos nós. Porque saber explicar não é a mesma coisa que saber o que é algo, digamos assim. Sim? Bom, é para ser, se não, seria, eles me diriam, passa pelos ativos e lá está.

Vamos ver, de forma rápida e resumida , o sistema de nomes de domínio, ou DNS, como o conhecemos pela sigla em inglês, é aquela aplicação que foi criada para traduzir nomes em endereços, nomes em identificadores únicos da Internet. Os identificadores únicos para dispositivos de Internet são os endereços IP, que não são os únicos identificadores nem são únicos, mas bem, é a principal característica que um dispositivo que se conecta à Internet deve ter: um endereço, um identificador que o diferencia de todos os outros dispositivos, e esse é o endereço IP. Sim?

E nomes de domínio são como o equivalente aos nomes das pessoas na lista telefônica. Sim? O DNS atua como um diretório telefônico para a Internet. Traduzir nomes de domínio em endereços. Por quê? Porque são mais fáceis de lembrar do

que números. Sim, justamente por isso. E um monte de outras vantagens, muitas delas técnicas e operacionais, que vieram depois com isso, mas bem, a principal motivação para ter um DNS é traduzir nomes em endereços.

Então, agora vamos falar um pouco sobre como criar um sistema que originalmente traduzisse nomes que só podiam ser escritos em ASCII, para algo que pudesse ser virtualmente escrito em qualquer tipo de linguagem, digamos, codificado de uma maneira muito especial, que agora o Carlos vai me ajudar a tentar explicar, digamos, certo? Porque em UTF-8 é tipo, pode levar uma semana para entender. Bom. E fale um pouco sobre como isso é transferido para... para o e-mail.

Então, com aceitação universal, agora será possível ter, não, não agora, alguns anos atrás, era possível ter nomes de domínio e endereços de e-mail em idiomas, qualquer idioma, certo? O que você tem que fazer é criar algo como o dicionário, digamos assim, certo? A codificação para poder dar suporte a isso. Você quer que eu ... agora ou devo interrompê- lo ?

Carlos Martinez: Vá em frente. Bem, o problema é este. Todo mundo ouviu... Não estou dizendo que... Não estou dizendo que eles sabem o que é. Você já ouviu falar de UTF-8? Ninguém? Você? Você também? Estou olhando para você. Bom, aqui está a questão. Você sabe o que é ASCII? Ou parece familiar? Parece mais familiar para você? Bem.

Em suma, qualquer sistema de comunicação digital que se conecte, digamos, que se comunique com zeros e uns, e não estou me referindo necessariamente a computadores, na verdade o ASCII vem do TELEX, a origem do ASCII vem do TELEX, nos anos 60, 70, 50, é uma forma de representar caracteres em informações digitais. A primeira versão do ASCII tinha 7 bits, nem mesmo 8. Por que 7? Porque não havia nem mesmo a noção de que 8 fosse um tamanho razoável para um computador digital. E naquela época cada bit, não cada byte, cada bit valia muito dinheiro. Transmitir um pouco corretamente a longa distância custa muito... custa muito dinheiro.

Então, isso... Uma agência nos Estados Unidos, acho que foi a antecessora da FCC em alguns aspectos, disse: bem, precisamos de uma maneira padrão de representar caracteres para que possamos enviá-los via TELEX. E eles decidiram por 7 bits porque em 7 bits, em 128 combinações, eles disseram, aqui podemos codificar tudo o que precisamos para alcançar uma comunicação inteligível.

Nico disse algo muito bem: as pessoas presumem que o ASCII representa o inglês, mas não é. ASCII representa o menor subconjunto do inglês que permite que algo seja transmitido de forma mais ou menos compreendida. Se você pesquisar no Google, por exemplo, por imagens de telegramas antigos, mesmo em inglês, verá que nem todos os caracteres foram transmitidos. Muitas vezes as vogais foram omitidas. Nas palavras em que se entendeu o mesmo, as vogais foram omitidas. Cada carta custava muito dinheiro. Então, bem, essa parte... Não é como se eles tivessem tirado, nunca lhes ocorreu colocar. Eles disseram: qual é o mínimo, mínimo, mínimo que precisamos? 7. Qual é o mínimo, mínimo, mínimo que precisamos? Porque além dos caracteres, letras minúsculas, números, pontuação, havia certas sequências de controle. Quer dizer, tudo estava indo. Na verdade, havia uma faixa de bits que era transmitida e tudo tinha que ir lá, incluindo as informações de controle, onde a mensagem começa, onde a mensagem termina, etc. Ótimo.

Com o advento dos computadores digitais, houve um período de transição em que as pessoas não conseguiam concordar sobre o que era um agrupamento razoável de bits. Havia arquiteturas muito estranhas, 13 bits, 17 bits, 36 bits, coisas assim. E em algum momento as pessoas chegaram à conclusão de que 8 era um bom número. Então, é aí que entra o conceito de byte. Só então o conceito de byte aparece.

Alguns anos se passam e as pessoas dizem, bem, esse ASCII que estamos usando, mas o usamos porque era a única coisa que havia, não poderíamos estendê-lo para 8 bits e usar esse bit que não estamos usando para alguma coisa? Bem. E aí surge o que você provavelmente já conhece, o ASCII que todos vocês conhecem, que é um pouco mais rico que o ASCII original.

Agora, os anos passam e a computação pessoal começa a se tornar popular, e aí vem o Windows. E no Windows diz Microsoft, seria legal poder tornar isso multilíngue , porque temos clientes no mundo todo, mas não tem como representar esses caracteres. E então eles inventaram um hack , literalmente um hack , que nós sofremos até agora em certos aplicativos, que eram o que eles chamavam de Páginas de Código do Windows . Sim?

Então o que eles fizeram foi manter os 7 bits do ASCII como se fossem ASCII. Mas dependendo, não me lembro qual valor, os próximos 128, ou seja, os 128 que você ganhou adicionando um bit, eram usados para representar caracteres em outras línguas. Então, por exemplo, em espanhol e outras línguas latinas, eles eram, eu acho, CP 1259, algo assim. Mas se você fosse para a Rússia, a história seria outra. E

na China foi outro, etc., etc. E, ao mesmo tempo, os 128 que você ganhou não foram suficientes para representar idiomas, por exemplo , o chinês, que tem muito mais símbolos. Claramente isso é um problema, e também houve um esforço de padronização, que levou a algo chamado Unicode, cujo nome está escrito em algum lugar, eu acho, no slide anterior .

O Unicode visa continuar expandindo essa ideia, representando caracteres ou coisas mais genéricas, como emojis, por exemplo, bonecos. Tenho minha opinião muito pessoal sobre isso. Não tenho necessidade disso. Mas bem, continue representando essas coisas não como um byte, porque um byte não é suficiente, mas como consequências de um byte. Mas sempre mantendo aquele critério de eficiência que mencionei antes.

Então, uma aproximação simplista seria: bem, se um byte não for suficiente, colocarei dois, ou três, ou quatro. O problema é que se cada caractere que eu uso ocupa quatro bytes, as coisas de repente ficam enormes. Isso pode ser mais ou menos um problema agora, mas o problema é que quando o Unicode foi pensado pela primeira vez, o armazenamento ainda era uma das coisas caras, assim como o custo de transmissão. O custo da transmissão, o custo da transmissão, vai voltar a ser um tópico, sobre o qual falarei mais tarde.

A questão é esta, então essas pessoas do Unicode disseram: bem, quanto espaço precisamos para representar tudo? Ou seja, todos os caracteres de todas as línguas conhecidas, mais o emoji de rosto sorridente, e o emoji de mate, e o que eu sei, e eles chegaram a um número que não sei como chegaram, mas é um número meio estranho, é um milhão 111 mil, não sei quanto, que na realidade em Hexa é escrito como 10 FFFF, esse é o máximo.

Então, bem, lá você pode baixar o banco de dados Unicode, que contém todos os mapeamentos dessas coisas. Ler algumas partes é realmente interessante, porque cada personagem vem com o que chamamos de código. ponto , que é o número que o representa, uma descrição no texto, por exemplo, o ñ diz n latino com til, essa é a descrição que eles têm em inglês para o ñ, e assim por diante, tudo, tudo, tudo, tudo, tudo, foi quando eu descobri isso, que existem caracteres largos e estreitos em alguns scripts, e eles significam coisas diferentes, ou seja, o mesmo caractere, largo e estreito, significa coisas diferentes.

Então, além disso, uma vez que temos esses personagens representados, surge a necessidade de transmiti-los através de um canal de comunicação. A aproximação mais simplista seria: quantos bits eu preciso para representar esse estranho

número de um milhão, não sei quanto, e mais ou menos você tem o suficiente, e vai ficar tudo bem, você tem que expressar 4 bytes. Como isso é extremamente ineficiente, foi criado o conceito do que eles chamam de regras de codificação, que é como representar esse número de 32 bits, da forma mais eficiente possível, em bytes, para transmissão e armazenamento.

E é aí que entram os UTFs , existem vários UTFs . Transformação Unicode Formato , há UTF, Transformação Unicode Formato . Então, tem o mais comum, que é o usado no DNS, no e-mail, que é o UTF-8. Sim? Não é porque é, todo mundo quer que seja, porque 8S tem a ver com o byte, não tem realmente a ver com isso, é simplesmente, na verdade , uma codificação de comprimento variável.

Em UTF-8, caracteres ASCII, que já existiam no ASCII original, são representados como seu valor ASCII. E os outros, os raros, estão codificados ali, dependendo de onde estão, nessa faixa de 0 a 1.100.000, não sei quanto, podem ser, podem ser codificações de 1, 2 e 3, não sei se são 4, talvez sejam 4 bytes também. E há algumas partes no meio que são usadas para dizer, bem, o que vem a seguir, seja um novo personagem ou uma parte daquele que estou montando. Então esse é o famoso UTF-8.

Existe o UTF-7, que está obsoleto, não é mais usado, ou não deveria mais ser usado, e sempre temos algumas surpresas aí. Quando você deprecia algo, pode ter certeza de que em 35 anos você vai abrir algo, e isso vai estar codificado naquilo. E o que é usado, em algumas aplicações, é algo chamado UTF-32, que é uma codificação muito simplista, que é só isso, pegar cada caractere, e então, a vantagem disso, aqueles que já tiveram dificuldades com programação, sabem que processar coisas de comprimento fixo é geralmente muito mais fácil do que processar coisas de comprimento variável. Portanto, em certas aplicações, o UTF-32 é o preferido.

Então, bem, esse é o primeiro, como posso dizer, o primeiro segredo obscuro de tudo isso que estou aqui para contar a vocês. Aí está.

Nicolás Antoniello : E continuando um pouco com o que o Carlos estava dizendo, gostei do exemplo dos emojis, é um exemplo muito bom para quem sabe mais, está mais acostumado com... Você sabe que em muitas aplicações, digamos, para colocar uma carinha de riso, você pode escrever, sei lá, dois pontos, rsrs, e isso significa rir. fora alto , ou rindo muito, digamos, e dois pontos novamente. Então, eles colocam isso entre dois pontos, e o que quer que esteja entre eles, o aplicativo interpreta como, ah, eu tenho que codificar isso, é uma codificação, não é que eu

queira transmitir dois pontos, rs, dois pontos, o que não significa nada, mas é uma carinha sorridente. Então ele substitui por um rosto sorridente.

No DNS é a mesma coisa, digamos assim. Através do UTF-8, e uma codificação que para o DNS, por questão de aceitação universal, é chamada de Puny O código é escrito usando caracteres ASCII, algo que os aplicativos e sistemas interpretam como o que deveria ser, que é um caractere internacionalizado, ou seja, internacionalizado significa qualquer coisa que não seja ASCII, digamos assim, certo?

Isso, a verdade é que eu estou pensando que é um pouco, isso, tipo, não importa, internacionalizado quando não é ASCII, é como quando a gente fala, vou para o interior, e de repente eu vou para Piriápolis, Piriápolis não é mais lá, não tem mais nada lá fora além de Piriápolis, porque é no litoral, mas pra gente é o interior, né? Bem, é a mesma coisa. Internacionalizado significa que não é ASCII, tudo bem, coloque .

Bem, então, e é assim que está codificado, no DNS e é assim que está codificado no , no , no e-mail, e é um pouco, um pouco também o que, o que esse formato MIME faz, que, que permite que você pegue uma imagem e a converta em texto e a transmita como texto e então, do outro lado, ela é reconstruída e se torna uma imagem, digamos, certo? Então, para enviar uma pequena imagem como essa, eu gero 57 folhas de MIME, de texto, e então isso, de novo, isso, de uma forma absolutamente ineficiente, a menos que eu compacte, eu traduzo de volta para a imagem.

Então, falando um pouco sobre e-mail e nos aprofundando mais no assunto, só para revisar, essas são três grandes partes de um e-mail: o que seria o envelope, o cabeçalho da mensagem e o corpo da mensagem, digamos assim. O, o e-mail, quando, quando eles o fizeram, era em formato eletrônico, correio postal tradicional, certo? Então, este, o, o envelope virtual, no caso de e-mail, tem um remetente e um destinatário do lado de fora, digamos. Esses, esses são, esses são parte dos cabeçalhos das mensagens, digamos, e são usados pelo sistema de envio e, e, como você diz isso? Entrega , seria tipo, sim, tipo envio. O sistema de envio de e-mail usa esses cabeçalhos e essas informações para criar o e-mail, certo? Por exemplo, de um cliente de e-mail onde eu escrevo um e-mail para a caixa de correio do destinatário, passando por vários servidores e dispositivos no meio.

E o corpo da mensagem é o que vai dentro do envelope, certo? Estou lhe dizendo que aceitação universal não significa entrar no corpo da mensagem. Se o corpo da

mensagem estiver escrito em qualquer idioma em que esteja escrito, o, o, o, o projeto de aceitação universal não se importa, porque é isso que vai dentro do envelope. Assim como o carteiro não se importa com o que tem dentro do envelope, certo?

Para ser justo, isso já foi um problema. Mas isso foi resolvido bem antes, quando, em geral, quase todos os sistemas operacionais, mais ou menos em uma janela de tempo por volta do final dos anos 90, adotaram o UTF-8. Todos os sistemas operacionais se tornaram UTF-8. Então, o que vai dentro de um e-mail, de um documento do Word ou mesmo de um arquivo de texto não é mais um problema. Nomes de arquivos, por exemplo, foram um problema por muito tempo e, bem, continuam sendo um problema até hoje.

O que não significa que não seja um problema por questões de segurança, etc., etc. Mas, ei, é a mesma coisa. Eu posso enviar uma bomba em um envelope e posso enviar um arquivo compactado em um e-mail que é uma bomba digital, digamos assim, certo? Algum tipo de ataque, vírus Trojan. Hoje em dia, há uma infinidade de danos que podem ser causados por um e-mail, certo? Mas, para fins de aceitação universal, não vamos nos concentrar no que está dentro, mas sim no que está fora do envelope. Sim?

Bom. Essas são definições que, quando você cruzar a porta, se você esquecê-las, sua vida será melhor. Eles são como os diferentes dispositivos e agentes envolvidos na cadeia de envio, recebimento e transmissão de um e-mail, certo? Então, temos o MUA, que nada mais é do que uma sigla muito complicada para descrever o cliente de e-mail, que pode ser qualquer cliente de e-mail, não sei, Outlook, qual é o nome do da Apple? Não me lembro mais, mas bom, são muitos, o que você quiser.

Na época, aqui estou eu olhando para o Martín que trabalhava, em outra vida nós trabalhamos juntos em algum lugar, usávamos o Paint, você lembra do Paint? Um cliente de e-mail baseado em texto que era apenas para nerds, por assim dizer. Sim, sim, tudo foi feito com as chaves daquele jeito. Ei? Não foi o pior. E ele não era o pior, havia piores, havia piores, havia piores. Você estava chapado... Sim, sim, sim, você fazia as coisas muito mais rápido hoje em dia e para alguns, hum, eu queria ter continuado com isso, com isso... Para o A nostalgia da pintura continua existente o MUT. Bom, vamos usá-lo esta tarde para uma demonstração, então aí está.

Isso, o MSA, que já está no lado do servidor, digamos assim, isso, o e-mail, na verdade, no , no , faz parte do sistema que envia o e-mail que se gera no MUA,

digamos assim. O MSA é quem inicia o envio, o primeiro que envia o e-mail gerado, digamos. Então, os MTAs são todos, vamos ser diretos e rápidos, mas para deixar claro, todos os sistemas pelos quais os e-mails passam. Todos os servidores intermediários pelos quais um e-mail passa ao viajar da origem até o destino.

Então, você verá que, normalmente, haverá um MUA, um MSA, vários MTAs e um MDA no final, que é aquele que recebe o e-mail, e o MDA é o programa que pega o e-mail que chega e o coloca, o deixa na caixa de correio, o deposita na caixa de correio do usuário destinatário, digamos assim. Sim?

Bom. Por que estamos falando sobre tudo isso ? Aí ele apareceu, não era a IEA, mas a EAI. Bem. Tendo suporte a UTF, coisas que têm que acontecer, que o Carlos já explicou um pouco, é que você tem que ter suporte a UTF-8 para poder gerar essas codificações em linguagens diferentes de ASCII, tanto para o nome da caixa de correio quanto para o nome do domínio.

suporte a domínio universalizado, meu, todo o meu sistema DNS tem que suportar domínios internacionalizados e isso significa que não apenas meu servidor DNS onde eu coloco as informações que são chamadas de servidores autoritativos têm suporte para isso, mas os servidores recursivos também têm que ter , eu tenho muitos, todos os servidores que participam da resolução do nome de domínio têm que suportar isso.

E a mesma coisa, se eu quiser ter um nome de caixa de correio internacionalizado, mesmo que eu não tenha um domínio internacionalizado, por exemplo, nicolascontildela at internet.com.uy ou qualquer outro, vou precisar de todos os servidores que participam do e-mail, ou seja, toda essa família de servidores da qual estávamos falando, todos eles, onde quer que esse e-mail vá, para suportá-lo. Se o e-mail chegar a um servidor intermediário, um MTA intermediário, por exemplo, que não ofereça suporte a isso, ele estará inativo. É só isso. Sim?

E o que acontece depois disso é uma roleta. Existem alguns sistemas, esta é uma opinião pessoal , que são bem configurados e enviam uma resposta para que a origem saiba o motivo pelo qual não consigo prosseguir. Basicamente, quem rejeita isso dirá: não suporto a universalização de caixas de correio ou nomes de domínio, ou nenhum dos dois. O e-mail chegou a esse ponto, mas já aviso que é por isso que vou descartá-lo. Se eu descartei, tchau. E esse e-mail nunca chegará ao seu destino, digamos assim. Não importa quantas vezes você envie. Então toda a cadeia tem que dar suporte à aceitação universal. Esse é um dos obstáculos para desenvolvê-lo. Não depende só de mim, depende de tudo o que acontece no meio.

E bem, há algumas coisas que não são universalmente aceitas. Digamos que o UTF-8 já existia muito antes e era utilizado. É outra ferramenta que também foi utilizada para este tópico. O conteúdo do assunto do e-mail não é relevante para o envio do e-mail. Na verdade, isso não lhe interessa, interessa apenas ao remetente e ao destinatário. Comentários do diretor. Não vamos entrar em detalhes sobre isso, mas de qualquer forma, o endereço é repetido duas vezes. Ele sai do envelope para o carteiro usar e volta para dentro do envelope. O que vai dentro do envelope não importa. Quem está dentro do envelope pode até mentir sobre o que está dentro do envelope, mas o carteiro se importa com o que está fora do envelope e isso também é universalmente aceito. E obviamente o corpo da mensagem também não, porque ela vai dentro do envelope e eu posso escrevê-la em qualquer idioma que eu quiser.

E bem, para a universalização de tudo aquilo que não é exatamente de aceitação universal, já existe o MIME que, combinado com o UTF-8, permite representar quase tudo. Por exemplo, no corpo de uma mensagem de e-mail, até mesmo uma imagem, um documento de processador de texto, uma apresentação do PowerPoint, um PDF, qualquer coisa.

Bom. Então, entrando em mais detalhes e configuração, porque já estamos quase na hora, mas estamos cinco minutos à frente. Deixaremos alguns requisitos desta apresentação para que aqueles mais técnicos, aqueles mais curiosos, possam ter isso detalhado. Essas são algumas das questões que precisam ser levadas em conta quando estou configurando a aceitação universal ou quando quero fornecer suporte à aceitação universal.

Bom, isso é como uma pequena receita, você tem que normalizar a string Unicode antes de processá-la em UTF-8, usar o formato correto para IDN, e bem, existem vários formatos para muitos desses padrões que eram usados no começo, mas não são mais usados. Não é mais recomendado usá-los, mas sim usar sempre a versão mais recente, digamos, de todas essas coisas. Se, por exemplo, o IDN , muitas aplicações são programadas usando essa versão, versão 2003, essa versão não é mais recomendada para uso, então agora quem já tem isso programado terá que lançar um patch com essa versão, e quem está começando a programar não deve usar essa, deve usar essa versão do IDN.

Carlos Martínez: Você nos contou o que é IDNA? Não, diga a ele . Eles perceberam que essa coisa do Unicode permite que você faça muitas coisas e represente muitas coisas. Na verdade, isso permite que você faça e represente tantas coisas que, de certa forma, isso se torna um perigo. Então, para fins de uso na Internet,

para comunicações na Internet, nem toda a variação, a enorme variedade de coisas que o Unicode permite que você represente são permitidas. Portanto, há uma série de regras que determinam o que é válido e o que não é válido para enviar pela Internet. Que elas seriam válidas dentro do corpo da mensagem, um pouco como o que Nico estava dizendo, de repente.

O mais óbvio, por exemplo, é que emojis não são permitidos em domínios de nível superior . Quer dizer, nunca haverá, ou bem, enquanto estivermos vivos, nunca haverá, sobre o meu cadáver, nunca haverá um domínio de nível superior que seja ponto mate, digamos assim. Ponto fosco, desenho fosco, quero dizer. Esse é um exemplo engraçado, mas há muitas coisas assim. Porque há toda uma questão de segurança ligada a isso que talvez possamos discutir, que são os ataques homográficos. Isso é tremendo.

Quero dizer, a quantidade de coisas que você pode fazer e a quantidade de enganos que você pode cometer em um mundo onde as pessoas ligam para você e pedem dinheiro, e você dá a elas. Quer dizer, imagine as coisas que você pode fazer desenhando com personagens estranhos. É incrível. Então, bem, há uma série de regras que determinam quais coisas são válidas para representar em endereços de e-mail, nomes de domínio, especialmente nomes de domínio, que têm a ver com essas regras do IDNA. Além de alguns regras de codificação adicional e tal .

Nicolás Antoniello : Ótimo. Deixei a apresentação para o final porque veremos o restante do que está no meio da apresentação à tarde, enquanto fazemos o laboratório de demonstração. Faremos uma espécie de demonstração com Carlos Guiada à tarde, depois do almoço. E é aí que vamos fazer a apresentação.

Só queria deixar isso como um aviso para o caso de você ter um momento agora ou quando quiser. Esta é uma página disponível no site da Aceitação Universal . Direção Grupo , digamos, que pode ajudar você a verificar se seu servidor de e-mail suporta internacionalização de e-mail. Isto é, se basicamente apoiar a aceitação universal. Sim? E eles acessam lá. E basicamente, eles vão gerar um e-mail, vão enviá-lo, e o sistema vai ver se obtém uma resposta e depois vai ver qual o nível de resposta obtido.

Porque dentro do tópico de... de aceitação universal e suporte de sistemas... Desculpe, e-mail... Esse é o problema de comer... Ah, amido de milho antes... O e-mail tem vários níveis de suporte. E isso não significa que você tenha que obedecer a todas elas. Ou seja, você deve ter suporte absoluto e completo para poder transmitir um e-mail. Por exemplo, se for um MTA, que é um agente intermediário,

que não é quem gera o e-mail nem o destinatário e não oferece suporte à criação de uma caixa de correio universalizada... Bem, não importa, porque o MTA não é, na verdade, onde a caixa de correio é criada ou onde isso é manipulado, mas o MTA tem que receber um e-mail e repassá-lo. Seria como uma espécie de agência intermediária de correio.

Mas é preciso ter suporte para isso. Você precisa entender que é um caractere internacionalizado no nome de uma caixa de correio ou domínio para encaminhar e-mails ao seu destino. Então há níveis diferentes. Então, com essa ferramenta você pode verificar se eu consigo lidar com isso e em que nível eu consigo lidar com isso. E isso tem vários tipos. Tipo 1, tipo... Geralmente é suporte tipo 1, tipo 1 e tipo 2, digamos assim. O tipo 2 é aquele que suporta tudo. O Tipo 1 é um suporte parcial que pode ser usado, por exemplo, em um MTA, mas não pode ser usado para um agente de cliente de e-mail ou um destinatário de e-mail, digamos.

E bom, nós já falamos um pouco sobre isso no começo. Além do e-mail, a ideia é que... Tipo... Convidá-los para... Convidar a todos nós para isso, para revisar não apenas o software em si, mas no nível do aplicativo, no nível das interfaces do usuário da web, se estamos ou não apoiando a aceitação universal.

Na verdade, só o software não é suficiente, porque se esquecermos de todas as formas, por exemplo, para criar um endereço de e-mail, se a interface de... Hoje existem, por exemplo, todos os servidores DNS que são usados hoje, autoritativos e recursivos, em todas as suas versões, em todos os seus sabores, digamos, de qualquer fabricante, em versões de, não sei, cinco anos atrás até hoje, eles suportam a universalização de nomes de domínio.

Você ficaria surpreso com o número extremamente pequeno de registradores, agentes que comercializam nomes de domínio, que apoiam a aceitação universal, que apoiam a universalização. São muito, muito poucos. Todos os seus sistemas DNS já suportam isso. O que ele não suporta é a interface web, os sistemas de cobrança ou os sistemas de processamento. Então você também tem que fazer um investimento nisso, que eu acho que é o maior, porque o resto é atualizar o software e pronto.

Claro que, como tudo o mais, você tem que dizer uma coisa e fazer outra. Você tem que treinar as pessoas, tem que treinar as equipes de operações, etc. Mas a ideia dessas reuniões também é um pouco assim, para pelo menos dar o pontapé inicial para você começar a explorar isso. Bom, é isso por enquanto.

Moderador: Bem, muito obrigado, Carlos e Nicolás. Eu não contei a eles, mas Carlos e Nicolás são obviamente membros do capítulo, membros fundadores, e Carlos também foi presidente da ISOC-Uruguai, algo que esqueci de mencionar, mas bem, todos nós os conhecemos e a verdade é que eu disse a eles que eles eram uma boa dupla fazendo apresentações. Certo, agora voltando ao assunto, falando sobre a questão dos emojis que você mencionou, Carlos, há vários grupos de discussão na GNSO da ICANN sobre a inclusão de emojis.

Carlos Martínez: Tentador, quero dizer, abre muitas oportunidades de negócios. Muitos. Não é algo muito genuíno, mas, enfim, são negócios. Mas sim, certamente é um tópico de discussão, mas também cria um mundo de problemas gigantescos.

Moderador: Sim, houve justamente aqueles a favor, aqueles contra, várias discussões muito interessantes sobre isso.